Benchmarks for IP Forwarding Tables

M. Castelino, R. Gunturi
Intel Corporation,
Santa Clara, CA, USA
manohar.r.castelino@intel.com
ravi.gunturi@intel.com

V. Filauro, G. Vlantis
STMicroelectronics,
San Jose, CA, USA
valerio.filauro@st.com
george.vlantis@st.com

M. Campmas, A. Coppola
Cypress Semiconductor.
San Jose, CA, USA
mjc@cypress.com
aco@cypress.com

Abstract

IPv4 and IPv6 route tables are ever changing and a subject of constant study. The Network Processing Forum (NPF) required a parameterized selection of IPv4/IPv6 route tables, real and synthetic, to enable benchmarking the performance of systems that include Network processors and search engines (co-processors). The benchmarks have scalable parameters, and are to be run in a standard way, as specified in the NPF IP Forwarding Implementation Agreement. A synthetic traffic generator was also required to implement the benchmarks.

Requirements were collected, and a freely available software toolkit was created. The kit has two sets of components – one set which generates IPv4 forwarding tables in the range of 1000 to 1 million prefixes, along with IPv6 tables of size 400 to 1200 prefixes, and another set which generates synthetic traffic patterns for the benchmarks. The toolkit's programs can be used for NPF benchmark certification, or as a suite of helpful table and traffic generation tools.

This paper explains the empirical schemes devised to create IPv4 and IPv6 route tables in line with these requirements, along with the algorithms used to implement the schemes. It also describes the problem of generating traffic to exercise the route tables produced using the previous schemes. The results obtained by running this software to generate route tables are presented and evaluated.

Keywords

IPv4, IPv6, traffic, forwarding table, benchmark

1 Introduction

1.1 Motivation

From 2002 to the early part of 2003, the Benchmark Working Group in the Network Processing Forum (NPF) drafted a benchmark implementation agreement for IP Forwarding to address the demand for a methodology to benchmark IPv4 and IPv6 operating on a network processing (NP) system. The NPF had already developed an IPv4 Forwarding Benchmark Specification, which served as a starting point for this Benchmark Specification.

The IPv4 Benchmark Specification used the Mae-West route table, containing about 28000 routes, as the reference route table for the benchmark. We received several comments from NPF members, and system vendors, who reviewed the specification that this route table was too small and not representative of route tables on current core routers. Furthermore, a reference route table had to be developed to benchmark IPv6 performance on a network processing system. The NPF Benchmark Working Group realized the need for a set of reference route tables representative of those seen at current routers for the IP Benchmark Specification.

1.2 Process

A study group in the NPF investigated IPv4 and IPv6 route tables to decide on a set of reference route tables for use in the IP Forwarding Benchmark that has since become an NPF Specification.

Three individual problems had to be addressed. The first one was to generate or obtain large IPv4 route table(s) with distributions representative of large tables found at current routers on the Internet. The second problem was to generate or obtain representative IPv6 route table(s). The last issue was generating the right traffic to exercise the routes in these tables.

This paper describes the requirements and the empirical solutions used for the three problem areas identified by the study group – generating IPv4 route tables, generating IPv6 route tables, and creating traffic to exercise the generated tables. A full report of our study, the route table implementation kit and the traffic generation kit are bundled with the IP Benchmark Specification Implementation Agreement and available directly from the

0-7803-8396-6/04/$20.00 © 2004 IEEE

NPF website (http://www.npforum.org). The rest of the paper is organized as follows. Sections 2.1 and 2.2 present requirements for generating IPv4 route tables and the empirical scheme devised satisfying these requirements. Sections 3.1 and 3.2 similarly present requirements and the scheme for generating IPv6 route tables. Section 4 tackles the problem of generating traffic from a traffic tester to exercise routes in the reference route tables. Detailed results and analysis showing the efficacy of our solutions are presented in Section 5. The last section draws conclusions from our study and outlines areas where our work can be further expanded.

2 IPv4 Route Tables

2.1 Survey and Problem

The first of the three major challenges that the NPF Routing Table Study Group confronted was to identify, and if necessary, generate large IPv4 routing tables to be incorporated in the forwarding benchmark. In order to determine the desired number of entries within the large IPv4 routing tables, the study group surveyed several members companies of the NPF. In particular, system manufacturers and silicon vendors who market forwarding accelerators desired such a benchmark. Due to these wide variants in size requirements, the challenge for the Routing Table Study Group was to devise a method to generate routing tables that would scale up to one million prefixes, and scale down to arbitrarily small sizes.

Several methods were considered to generate routing tables with up to one million entries, before selecting the approach that was deemed the best. The idea behind the chosen approach was to maintain the distribution and structure of a reference "real-world" BGP routing table, and grow the table by selecting the most populated sub-branches first and replicating them wherever there were vacancies in the original distribution. The complete algorithm is described in detail in the next section of this paper.

A snapshot of the Telstra Autonomous System AS1221 Routing Table on 27 January 2003 was used as the reference BGP table to base the synthesizer. The Telstra database was chosen because it was the largest of the three (ASmap, MaeWest and Telstra) tables surveyed, and with approximately 135,000 entries was considered the best reflection of the current state of the Internet topology. Other sources of BGP tables were surveyed [31], but none were found that were more representative than Telstra AS1221.

Finally, a sub-sampling mechanism was chosen for generating smaller routing tables, so that system vendors can benchmark applications where tables that are smaller than the "real-world" BGP routing tables are desirable, or where the limitations of commercially available network traffic generators comes into play. This mechanism is presented in the section on Traffic Generation in Section 4

2.2 IPv4 Synthesis Algorithm

The goal of this algorithm is to generate a single larger forwarding table (with at least 1 million prefixes) preserving the following properties of a reference real world route table taken as input.

- Prefix Length Distribution (PLD)
- Prefix Depth Distribution (PDD)
- Prefix Height Distribution (PHD)

This terminology is defined in [7].

Also, the fraction of redundant prefixes in the larger table must be comparable to the fraction of redundant prefixes in the reference real world route table.

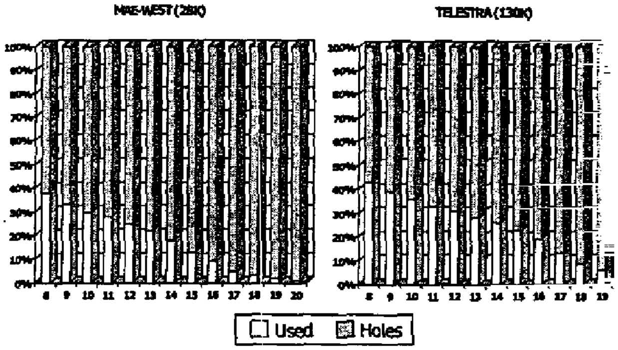

Figure 1 - Percentage of used values at different prefix depths for (a) Mae-West and (b) Telstra

The inspiration for this algorithm came from constructing a binary tree of the IPv4 prefix route table. A key observation from this construction is that the IPv4 prefix table is composed of a collection of sub-trees spread out (non-uniformly) across the IPv4 address space. Figure 1(a, b) illustrates the percentage of used values for different prefix depths, for both Mae-West and Telstra route tables. For example, Figure 1a shows that only 98 out of a possible 256 subnet IDs are used at prefix depth 8 (i.e., /8) in the Mae-West routing table. Therefore, there are 158 holes, or absent prefixes, at this level. Similarly, Figure 1b reveals 656 holes at prefix level 10 in the Telstra routing table [29].

Based on this observation, the devised scheme fills up the holes at each prefix depth in the tree. The holes are filled by replicating all sub-branches of the already used prefixes at that depth in a round robin fashion until the desired number of prefixes is achieved. Only the Unicast address space is used while filling up the holes. Reserved IPv4 addresses such as restricted or multicast IPv4 addresses, which make up about 40% of IPv4 address space, are not used.

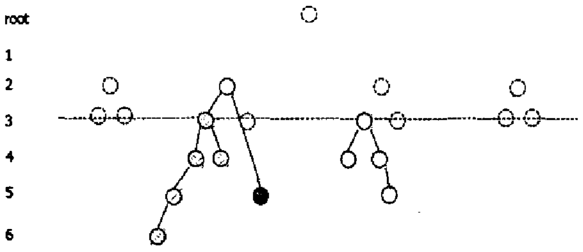

Figure 2 - Sample binary prefix tree of reference

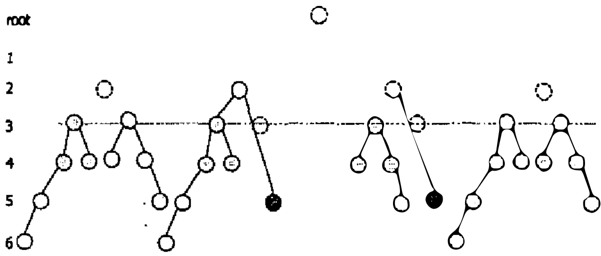

Figure 3 - Above example after replication scheme

Figure 2 and Figure 3 illustrate the method to fill holes. Figure 2 shows part of a sample binary tree representation of a reference route table. Figure 3 shows the same binary tree after the replication scheme discussed above has been applied to prefixes of length 3.

The pseudo-code for the replication scheme is shown in Figure 4. The presented scheme implies that this method of replication is applied to prefixes of every length. However, it is possible to modify the scheme to perform prefix replication only for selected prefix lengths. For example, since most IPv4 prefixes fall on class (class A, class B, class C) boundaries (/8, /16, /24), one could choose to replicate prefixes only at these points in the tree. Another possibility might be to start at prefixes of length 8 and choose only those lengths which have the most number of prefixes associated with them. The value of the lengths would vary with the choice of reference route tables. The number of prefix lengths used will depend on the amount of enlargement required.

The above scheme was implemented in [29] and used to enlarge the Mae-West routing table to 260K routes and the Telstra routing table to about 1 million routes. The algorithm started with /8 prefix lengths, and used consecutive prefix length values (/9, /10 etc.) until the desired forwarding table size was obtained.

An analysis of these enlarged tables showed that the prefix height, depth and length distributions are well preserved for the enlarged version of both the Mae-West and Telstra route tables with respect to the original real world route tables. The fraction of redundant prefixes in the enlarged route tables was also within the acceptable range. A complete set of results can be found in Section 5.

We present below an intuitive argument explaining why these characteristics are preserved by the scheme. The scheme works by replicating the structure of currently allocated prefixes in the prefix tree onto the unused sections in the tree. For example, look at the replication scheme shown in Figure 2 and Figure 3. The size of the enlarged prefix table triples after applying the replication scheme. From the figure, however, it is also evident that the number of prefixes of a given length has also tripled. Similarly, since the holes have been filled by mimicking the existing structure across the unused prefixes of a particular prefix length, the number of subset nodes (i.e., prefix height) for a particular superset node remains the same. Similarly, the prefix depth also remains the same. Of course in an actual tree, the structure may not be replicated exactly, since standalone prefixes or other prefix trees may already exist at that location in the tree. However, this greedy heuristic approximates the structure of the original prefix table well.

One of the limitations (or features) of this scheme is its lack of adaptability or predictability. The scheme generates an enlarged route table with essentially the very same properties of the reference route table. It assumes that a larger prefix table will have the exact same characteristics of the reference route table and does not attempt to adapt or predict the future characteristics of a large route table. However, this is not quite a limitation since the requirements also stipulated that the larger route table mirror the characteristics of the reference route table. This scheme was chosen because it met the requirements outlined in 2.2, and employed a heuristic that preserved the characteristics of the reference route table.

```
    R(U_j) : all entries associated with U_j or a subset of U_j
N : number of (MS)Bits used for Subnet ID
    u : number of used Subnet ID values $(h+u=2^N)$
    U_j : used Subnet ID value, $j=[1,u]$
    ¬ : bit concatenation operation
    h : number of Subnet ID holes $(h+u=2^N)$
    H_i : hole Subnet ID value, $i=[1,h]$
    E : a generic entry of the routing table
    E_0 : the entry E with the first N bits stripped
```

```
    pick an N
        for each i in [1,h]
            select the next most populated U_j
                for each entry E in R(U_j)
                        create a new entry $E'=H_i ¬ E_0$
    repeat until desired number of entries is
reached
```

If an adaptive or predictive behavior is required in generating larger route tables, other analytical methods predicting Internet growth similar to the one proposed in [6], [28] can be derived and used.

3 IPv6 Route Tables

The NPF conducted a survey of the state of the IPv6 network, and an analysis of the distribution of prefix table at a particular autonomous system (AS), AS 4554 in December 2002. This survey and analysis were used to generate requirements for IPv6 route table sizes for NPF Benchmark Implementation Agreements. A detailed survey and quantitative analysis of AS4554 are presented in [29].

3.1 Requirements

IPv6 networks are still in its infancy, so the task of formulating requirements for future IPv6 tables was difficult. Unlike IPv4 prefix table sizes, there was no input from system vendors or customers with respect to IPv6 tables.

The best solution would have two IPv6 route tables – one route table that models the prefix distribution and numbering at the core of the IPv6 networks (/24s - /48s) and another that models the access network (/48 - /64) before data enters an organization's internal network.

However, the IPv6 network is evolving rapidly and will undergo many changes before the topology stabilizes. Therefore, we decided that only one IPv6 table would be generated. AS4554 was chosen as the reference route table because it was one of the larger tables available, and had a representative set of prefixes of varying lengths.

AS4554 has about 400 prefixes. Since there were no strict requirements about the size of the IPv6 tables, we decided to generate a table 3 times as big – 1200 entries. The 1200 number was an educated guess. We assumed that IPv6 prefix tables would grow over time. However, IPv6 allocates addresses in a hierarchical fashion, and we assumed that table sizes would not grow as drastically as IPv4 because of this aggressive policy of maintaining hierarchy.

We also realized that these requirements would have to be re-evaluated and likely changed within the next year or two as IPv6 network deployment became more prevalent and we obtained more route table and network samples.

3.2 IPv6 Table Generation Scheme

The IPv4 table generation scheme presented in Section 2.2 was used to generate larger IPv6 tables as well. This scheme preserves route table distributions well, and in the absence of specific information about the distribution of future IPv6 tables, the approach to mirror the current distribution was the most reasonable way to generate a larger IPv6 prefix table. Currently the scheme generates only Aggregatable Global Unicast addresses (2000::/3), as these are the only type delegated by the IANA at the time of writing, and always preserves the 16 high-order bits (formerly known as Top-Level Aggregation Identifier TLA) of the original entries [17], [26].

4 Traffic Generation

Having decided on schemes to generate route tables, the NPF conducted a study on traffic testers that would send in traffic to exercise these routes. It was clear from our study that traffic testers would not be able to exercise an arbitrary number of routes, and that we had to devise a method, feasible on a traffic tester, to exercise a representative set of routes in the prefix tables used for testing IPv4 and IPv6 performance of a network processing system.

4.1 Traffic Tester Survey

The conclusions presented here were made by investigating the Adtech, IXIA and Smartbits traffic testers.

Current traffic testers allow users to setup a limited number of streams (ranging from 255 to 4096), or packet sources per test interface. Each stream can be configured to transmit packets with a unique link layer and IP address, a key requirement for traffic generation. Traffic testers allow users to setup flows to transmit packets with a certain aggregate of IP addresses. However, the set of IP addresses that can be aggregated is restricted by the testers. Only subnets of an IP address may be aggregated into a flow. This is too restrictive if we wish to exercise a majority of the routes in the large IPv4 and IPv6 route tables created using the schemes discussed above.

4.2 Requirements and Features

The scheme chosen, had to be a method to select a subset of the route table and to generate traffic to exercise prefixes in that subset such that it was Uniform. The route tables would be uniformly partitioned over all interfaces, and the traffic generated for each interface would exercise a route in the partition corresponding to the incoming interface of the traffic. The distribution of routes exercised by the traffic on an interface would follow the distribution of routes in the routing table. The traffic generation also varied the traffic pattern to test for locality. Some vendors may have caches or small amounts of fast memory in their system, the benefits of which would be realized only in the presence of traffic exhibiting a high degree of locality.

4.3 Scheme to Create Traffic Streams

The algorithm is used generates a specified number of subsets of routes and destination addresses from an input IP route/prefix table. The algorithm has the following steps:

- **Route Table statistics collection,** which collects the prefix percentages and subnet distribution of the entire table.
- **Route Table normalization,** eliminates prefixes and routes in route table that cannot be distributed across all the interfaces.
- **Route distribution,** which extracts the specified number of the routes from the normalized prefix table such that they preserve the prefix length as well as subnet distribution found in the input table
- **Destination IPv4 address selection,** where IP addresses are selected to uniquely match each route selected. This step is required only while generating traffic to exercise entries in the shrunk route tables.

4.4 Analysis and Implementation

The traffic stream scheme was implemented [29] and tested with the Telstra reference route table. The next section shows the results obtained from these tests and presents further analysis. When the traffic stream algorithm is applied to a generated table, as described in Sections 2 and 3, the results show that the procedures preserve prefix length distribution approximately – only those prefix lengths that make up a sufficient percentage of the route table make it into the subsets. The normalization step will eliminate prefixes with specific prefix lengths, which cannot be captured in all the subsets.

4.5 Adjusting traffic locality

Traffic testers allow a stream to send a burst of packets at a time, instead of a single packet at a time. Thus traffic with a high degree of locality may be sent by configuring streams to send a large burst of packets instead of a single packet. The NPF IP Implementation Agreement specifies the burst parameters that must be used when testing performance of traffic with high degree of locality. The script to generate subsets takes in this burst size as a parameter when generating subsets and setting up traffic streams for test.

5 Software Package Description

The software implementing the various table creation algorithms and traffic generation drivers using those tables is embodied in a software package called The NPF IP Benchmark Implementation Kit [29], which has three directories, labeled:

- **Synthetic_Real_World** – Contains C++ programs parser, synth_table and statistics generate and analyze the results, respectively, for large IPv4 and IPv6 table generation.
- **Subset_Real_World** – Contains Tcl program generateSubsets.tcl, which generates small tables for IPv4 and IPv6 table generation. The statistics program above is used to analyze the results.
- **Generate_Traffic** – Contains Tcl program ixip.tcl. which calls other Tcl programs to generate traffic for the Ixia tester, using the routing tables produced by the large and subset table generators.

This toolkit enables the generation of large (100k-1M) and small tables (1k-50k) of IPv4 prefixes, and IP traffic based on those tables. It also can generate large (1200) and small (50) IPv6 synthetic tables. All of the source code is included in the package, as are the test scripts and documentation.

The toolkit is freely available from the Network Processing Forum website [29]. The package contains documentation and instructions on how to use the table and traffic generators, and includes make files, demo scripts, and test scripts. The large table generator is qualified, using GCC under Cygwin and Linux, and using Microsoft Visual Studio C++ under Windows. The small table generator has been qualified under Cygwin and Linux. The traffic generator directory holds the traffic generator, which uses Tcl, and the publicly available "mpexpr" Tcl package. It is qualified for the Ixia tester. Shell scripts in sh are supplied, which encapsulate parameterized running and testing of the table and traffic generation components of the toolkit.

5.1 Performance Metrics

The expanded table and subset table generation algorithms have the goals of matching the prefix length, depth, and height distributions (PLD, PDD, PHD) closely enough to the original distribution to enable the use of the new artificial tables in benchmarking future IP Forwarding systems.

For the Prefix Length Distribution results, Exponential Least-Square-Fit trend lines are used to compare the Cumulative Frequency Distribution (CFD) obtained from the original Telstra table, and the larger synthetic tables, generated from the Telstra snapshot, with 250k, 500k, and 1M prefixes. See Figure 5.

Two function metrics are used to compare the data set measures computed in this paper. The main one is the MAX-norm between two functions, which is the maximum of the absolute values of the difference between the functional points of interest. The second one used, exponential fit trend line, is for comparing the PLDs' of the large generated IPv4 tables.

For the most part (PLD, PDD, PHD, Redundant Prefix Distribution), are compared by visually inspecting the families of curves produced by the original and synthetic distributions, and implicitly computing the MAX-norm between the curves of interest. If any two curves in a family differ by more than ~10%, that family needs to be considered individually, or the generation algorithm has to be modified. Using this type of measure, the generated IPv4 large table, IPv4 small table, and IPv6 tables only show problems with the closeness of the PLD distribution of the IPv4 large table generation, and the Redundant prefix distributions, as is seen in Figure 6. The PDD and PHD graphs for all three data sets show very good matching, using the MAX-norm metric. Further analysis is performed on the large IPv4 tables only, as the Redundant prefixes are very small in relative number.

The PLDs of the large IPv4 tables are compared using visual inspection of the CFDs, and an exponential trend line applied to the log of the relative CFDs. The plot of the cumulative prefix probability distributions, on a linear-log scale allows an analytical comparison of the large table distributions. When the slopes of the trend lines are equal, and if the lines intersect, then the trend lines are equal. Figure 5 shows only the telstra_orig (135k) and telstra_1000k (1M) exponential fit trend lines, as the other two trend lines fall in between the envelope of those two, so would only add to the graphical clutter in the diagram. Comparing the two trend lines of the CDF curves is best done by taking the difference in slopes, which in the case of Figure 5 is:

```
slope(telstra_1000k )  = 0.499, and
slope(telstra_orig)    = 0.4158
```

Slope difference = 0.0832 , which is small enough to give less than a 2x difference between the telstra_orig and telstra_1000k curves over the CFD region of interest, which is when Prefix Length >= 17.

For predictability reasons, it is also required that the envelope of curves formed by the cumulative distribution and least squares exponential fit curves form an increasing monotonic collection of curves when viewed across all of the generated tables. This allows predictability of the scalability of traffic generation results when using the larger tables.

5.2 IPv4 Large Table Generation

The original large forwarding table used is a snapshot of the Telstra (AS1221) table, which has ~135k prefixes, with all prefixes having length no more than 32. The generated tables derived from the Telstra table snapshot, for the purposes of this paper, have sizes 250k, 500k, and 1000k. The user can generate whatever size table needed by varying the input parameters to the table generation software module.

The Prefix Height and Prefix Depth Distributions of the generated tables are easily seen to follow that of the original tables, which is to be expected by the algorithm description. See Figure 6.

5.3 IPV4 Small Table Generation

The Mae-West and Telstra tables were taken as the original tables. The generated tables, at 1k and 10k were seen to follow the goals set out for the measurement of the generated distributions described in the above.

5.4 IPv6 Small and Large Table Generation

The real world IPv6 table AS2554, of size ~400 prefixes was used to generate large IPv6 tables of size ~1200 prefixes. The same table was also used to generate smaller tables. The same overall expansion and subsets algorithms were used to generate synthetic IPv6 tables. Very different implementations were needed, though, due to the differing characteristics of these tables. The IPv4 and IPv6 programs were integrated into one program, which is controlled by a command-line switch. The distributions of the larger IPv6 tables meet the MAX-norm requirements of less than ~10% change between the original and generated tables.

5.5 Traffic Generation Results

Tcl scripts generating synthetic network traffic, based on the tables generated by the methods of this paper, for network traffic testers (e.g. Ixia, Adtech) were run and validated.

6 Conclusions

This paper has described the construction, delivery, and results achieved by creating an IP Forwarding Table Toolkit to enable the benchmarking of existing and future networking systems consisting of network processors and network coprocessors. Algorithms were implemented to expand or shrink existing tables. Tools are supplied to enable the generation of network traffic from these synthetic tables to allow benchmarking the systems under test.

A limitation of the schemes presented in the paper is that they generate smaller and larger route tables mirroring the state of the current network. None of the schemes is futuristic, or attempts to generate route tables based on network growth or network configurations predicted by studies such as [6],[7],[9],[28]. For example, the research in [30] presents an approach to model future route tables, with the specific goal of studying the best data structures for representing route tables. The next step in our research would be to expand on the ideas in [30], by combining our empirical approach to generating route tables with an analytical approach predicting the structure of a future network, to generate a prefix table that may be found at a router sometime in the future.

7 Acknowledgements

We are grateful to Frank Hady of Intel Corporation and Serge Audenaert of Alcatel Corporation for providing

valuable insight during the ongoing discussions that resulted in creating and adopting these schemes in the NPF IP Route Table Implementation Kit.

8 References

[1] http://www.iana.org/assignments/ipv4-address-space

[2] BGP Peering Routers. http://bgp.potaroo.net

[3] http://moat.nlanr.net/Routing/rawdata

[4] http://www.merit.edu/~ipma

[5] http://www.bgp.potaroo.net/as1221

[6] A. Coppola, Synthetic Route Table Generation Proposal, NPF2002.680.00. (http://www.npforum.org)

[7] A. Broido, E. Nemeth, KC Claffy. Internet Expansion, Refinement and Churn. In European Transactions on Telecommunications, 2002.

[8] T. Bu and D. Towsley. On distinguishing between internet power law topology generators. In Proc. of IEEE INFOCOM, 2002.

[9] T. Bu, Lixin Gao, D. Towsley. On Routing Table Growth. In Proceedings of Globe Internet, 2002.

[10] B. Carpenter, C. Jung. Transmission of IPv6 over IPv4 Domains without Explicit Tunnels. RFC 2529, March 1999.

[11] B. Carpenter, K. Moore. Connection of IPv6 Domains via IPv4 Clouds. RFC 3056, February 2001.

[12] Joseph Davies. Introduction to IPv6. Microsoft Corporation, February 2002. www.microsoft.com/windows2000/docs/ipv6.doc

[13] S. Deering, R. Hinden. Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, December 1998.

[14] Gennari, J., et al., Templates for IEEE Proceedings, available at http://www.sheridanprinting.com/ieee/ieee.htm

[15] R. Gilligan, E. Nordmark. Transition Mechanisms for IPv6 Hosts and Routers. RFC 2893, August 2000.

[16] R. Hinden, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[17] R. Hinden, M. O'Dell, S. Deering, An IPv6 Aggregatable Global Unicast Address Format, RFC 2374, July 1998.

[18] R. Hinden, Proposed TLA and NLA Assignment Rule, RFC 2450, December 1998.

[19] R. Hinden, S. Deering, R. Fink, T. Hain. Initial IPv6 Sub-TLA ID Assignments. RFC 2928, September 2000.

[20] IAB/IESG Recommendations on IPv6 Address. IAB, IESG. RFC 3177, September 2001.

[21] Internet Protocol. RFC 791.

[22] 6Bone test bed for deployment of IPv6. http://www.6bone.net

[23] 6Bone Japan: http://www.v6.sfc.wide.ad.jp/

[24] IPv6 Exchanges: http://www.v6nap.net/.

[25] IPv6 BGP Perspective at AS4554. http://www.ep.net/bgp-ipv6/

[26] INTERNET-DRAFT Network Working Group RFC3587 - R. Hinden, S. Deering, E. Nordmark

[27] Internet Assigned Numbers Authority (IANA) http://www.iana.org.

[28] C. Jin, Q. Chen and S. Jamin "Inet: Internet Topology Generator." Technique Report CSE-RT-433-00, University of Michigan, EECS dept. 2000. http://topology.eecs.umich.edu/inet.

[29] NPF IP Benchmark Implementation Kit. Part of IP Benchmark Implementation Agreement. http://www.npforum.org/techinfo/approved.shtml#ias

[30] H. Narayan, R. Govindan, G. Varghese. The Impact of Address Allocation and Routing on the Structure and Implementation of Routing Tables, In ACM Sigcomm, 2003.

[31] The Oregon Route Views Project. http://www.routeviews.org/

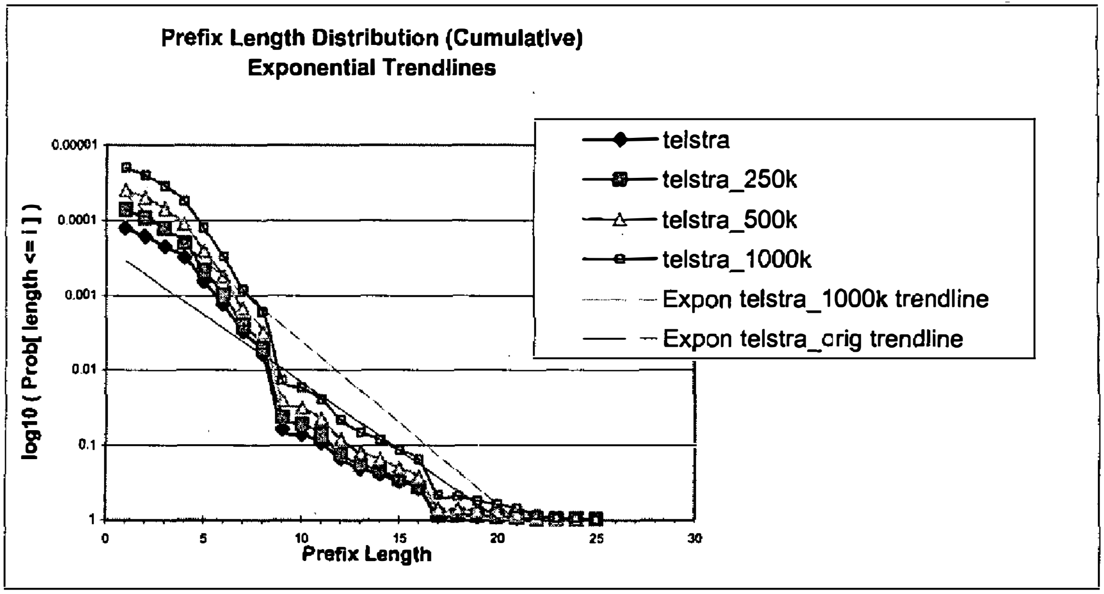

Figure 5: IPv4 Trendlines

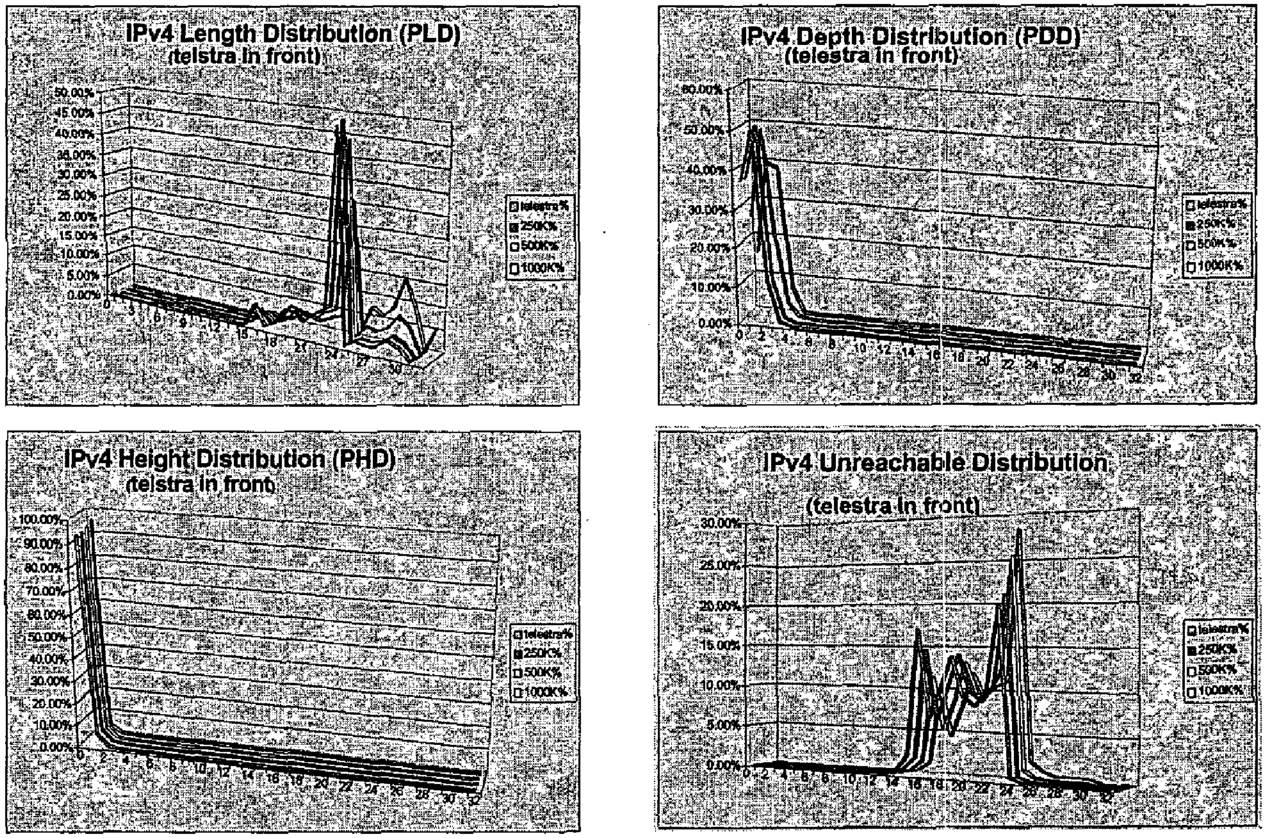

Figure 6 Results for route table expansion technique on IPv4 Telstra table